

What is claimed is:

1 1. A computer implemented method for distributing objects of a program, each
2 object containing one or more programmed member functions, said member
3 functions passing and/or returning one or more data stream objects as
4 parameters between at least two physical devices, said method comprising the
5 computer executable steps of:

6 identifying all of the objects in said program;

7 determining which of said objects are to reside on a first computer and
8 which of said objects are to reside on a second computer such that the
9 resulting distributed system comprises at least a first object on a first computer
10 and a second object and said one or more data stream objects on a second
11 computer;

12 identifying all programmed member functions that may be accessed from
13 a remote computer;

14 generating a first proxy and a second proxy for each said object that may
15 be accessed from a remote computer, said first proxy residing on said first
16 computer and said second proxy residing on said second computer, said first
17 proxy containing network linkage and indication to access programmed
18 member functions on said second proxy residing on said second computer

19 including logic to transfer said one or more data stream objects and said
20 second proxy containing linkage and indication to access said programmed
21 methods on said second object including logic to transfer said one or more
22 data stream objects; and,

23 accessing said remote programmed member functions through said first
24 and second proxies.

1 2. A computer implemented method as claimed in Claim 1 wherein:

2 -- said logic in said second proxy object in said second computer to
3 transfer one or more data stream objects to said first proxy object in said first
4 computer comprises the computer executable steps of:

5 creating or accepting one or more network connections with said
6 first proxy object in said first computer;

7 accessing each said data stream object to obtain the data each
8 said data stream object represents; and,

9 sending, via said one or more network connections, said data to
10 said first proxy residing on said first computer.

11 -- said logic in said first proxy object in said first computer to transfer one
12 or more data stream objects from said second proxy in said second computer
13 comprises the computer executable steps of:

14 accepting or creating said one or more network connections with
15 said second proxy;

16 creating a quantity of data stream objects equal to the quantity of
17 data stream objects to be transferred, each said created data stream object to
18 represent data received on said one or more network connections; and,

19 providing access reference to each of said created data stream
20 objects to said first object on said first computer.

21 3. A method as claimed in Claim 2, wherein data stream objects exist on both
22 said first computer and said second computer, and said method is performed
23 bidirectionally.

1 4. A method as claimed in Claim 2 or Claim 3, wherein one network
2 connection is created for each data stream object to be transferred.

1 5. A method as claimed in Claim 2 or Claim 3, wherein fewer network
2 connections than data stream objects are created, and said data is distributed
3 to said more than one data stream objects created on said first computer via
4 multiplexing.

1 6. A computer program product for distributing objects of a program, each
2 object containing one or more programmed member functions, said member
3 functions passing and/or returning one or more data stream objects as
4 parameters, across at least two physical devices, said computer program
5 product comprising:

6 a computer-readable storage medium having computer-readable
7 program code means embodied in said medium, said computer-readable
8 program code means comprising:

9 computer-readable program code means for identifying all of the
10 objects in said program;

11 computer-readable program code means for determining which of
12 said objects are to reside on a first computer and which of said objects are to
13 reside on a second computer such that the resulting distributed system
14 comprises at least a first object on a first computer and a second object and
15 said one or more data stream objects on a second computer;

16 computer-readable program code means for identifying all
17 programmed member functions that may be accessed from a remote
18 computer;

19 computer-readable program code means for generating a first
20 proxy and a second proxy for each said object that may be accessed from a
21 remote computer, said first proxy residing on said first computer and said
22 second proxy residing on said second computer, said first proxy containing
23 network linkage and indication to access programmed member functions on
24 said second proxy residing on said second computer including logic to transfer
25 said one or more data stream objects and said second proxy containing
26 linkage and indication to access said programmed methods on said second
27 object including logic to transfer said one or more data stream objects; and,

28 computer-readable program code means for accessing said remote
29 programmed member functions through said first and second proxies.

1 7. A computer program product according to Claim 6 wherein:

2 -- said logic in said second proxy object in said second computer to
3 transfer one or more data stream objects to said first proxy object in said first
4 computer comprises the computer executable steps of:

5 creating or accepting one or more network connections with said
6 first proxy object in said first computer;

7 accessing each said data stream object to obtain the data each
8 said data stream object represents; and,

9 sending, via said one or more network connections, said data to
10 said first proxy residing on said first computer.

11 -- said logic in said first proxy object in said first computer to transfer one
12 or more data stream objects from said second proxy in said second computer
13 comprises the computer executable steps of:

14 accepting or creating said one or more network connections with
15 said second proxy;

16 creating a quantity of data stream objects equal to the quantity of
17 data stream objects to be transferred, each said created data stream object to
18 represent data received on said one or more network connections; and,

19 providing access reference to each of said created data stream
20 objects to said first object on said first computer.

1 8. A computer program product as claimed in Claim 7, wherein data stream
2 objects exist on both said first computer and said second computer, and said
3 method is performed bidirectionally.

1 9. A computer program product as claimed in Claim 6 or Claim 7, wherein one
2 network connection is created for each data stream object to be transferred.
3

4 10. A computer program product as claimed in Claim 6 or Claim 7, wherein
5 fewer network connections than data stream objects are created, and said data
6 is distributed to said one or more data stream objects created on said first
7 computer via multiplexing.

1 11. A computer system for distributing objects of a program, each object
2 containing one or more programmed member functions, said member functions
3 passing one or more data stream objects as parameters, across more than
4 one physical device, said system comprising:

5 means for identifying all of the objects in the program;

6 means for determining which of the objects are to reside on a first
7 computer and which of the objects are to reside on a second computer such
8 that the resulting distributed system comprises at least a first object on a first
9 computer and a second object and said one or more data stream objects on a
10 second computer;

11 means for identifying all programmed member functions that may be
12 accessed from a remote computer;

13 means for generating a first proxy and a second proxy for each object
14 that may be accessed from a remote computer, said first proxy residing on said
15 first computer and said second proxy residing on said second computer, said
16 first proxy containing network linkage and indication to access programmed
17 member functions on said second proxy residing on said second computer
18 including logic to transfer said one or more data stream objects and said
19 second proxy containing linkage and indication to access said programmed

20 methods on said second object including logic to transfer said one or more
21 data stream objects; and,

22 means for accessing said remote programmed member functions
23 through said first and second proxies.

1 12. A system as claimed in Claim 11 wherein:

2 -- said logic in said second proxy object in said second computer
3 to transfer one or more data stream objects to said first proxy object in said
4 first computer comprises the computer executable steps of:

5 creating or accepting one or more network connections with said
6 first proxy object in said first computer;

7 accessing each said data stream object to obtain the data each
8 said data stream object represents; and,

9 sending, via said one or more network connections, said data to
10 said first proxy residing on said first computer.

11 -- said logic in said first proxy object in said first computer to transfer one
12 or more data stream objects from said second proxy in said second computer
13 comprises the computer executable steps of

14 accepting or creating said one or more network connections with
15 said second proxy;

16 creating a quantity of data stream objects equal to the quantity of
17 data stream objects to be transferred, each said created data stream object to
18 represent data received on said one or more network connections; and,

19 providing access reference to each of said created data stream objects
20 to said first object on said first computer.

1 13. A system as claimed in Claim 12 wherein data stream objects exist on both
2 said first computer and said second computer, and said method is performed
3 bidirectionally.

1 14. A system as claimed in Claim 11 or Claim 12, wherein one network
2 connection is created for each data stream object to be transferred.

1 15. A system as claimed in Claim 11 or Claim 12, wherein fewer network
2 connections than data stream objects are created, and said data is distributed
3 to said one or more data stream objects created on said first computer via
4 multiplexing.

5 16. A computer implemented method for distributing one or more objects of
6 a program across more than one physical device, each object containing one
7 or more programmed member functions, said member functions having

8 complex objects, said complex objects including one or more programmed
9 member functions, as parameters, said method comprising the computer
10 executable steps of:

11 identifying all of the objects in the program;

12 determining which of the objects are to reside on a first computer and
13 which of the objects are to reside on a second computer such that the
14 distributed system will consist of at least a first object on a first computer and
15 a second object on a second computer;

16 identifying all programmed methods contained in each object that may
17 be accessed from a remote computer;

18 generating a first proxy and a second proxy for each object that may be
19 accessed from a remote computer, said first proxy residing on said first
20 computer and said second proxy residing on said second computer, said first
21 proxy containing network linkage and indication to access programmed
22 member functions on said second proxy on said second computer including
23 logic to transfer and translate complex objects which reside on said first
24 computer used as member function parameters and said second proxy
25 containing linkage and indication to access said programmed member
26 functions on said second object including logic to transfer and translate
27 complex objects, said complex objects containing one or more programmed
28 member functions and reside on said first computer, used as member function
29 parameters; and,

30 accessing said remote programmed methods through said proxies.

1 17. A method as claimed in Claim 16, wherein:

2 said logic in said first proxy on said first computer to transfer and
3 translate complex data objects comprising the steps of:

4 creating a third proxy, for said complex object, which is to reside on said
5 first computer with said complex object, said third proxy containing linkage and
6 indication to access programmed member functions on said complex object;

7 creating a reference table entry which correlates said third proxy
8 object to said complex object, which may be accessed by said third proxy
9 object to access said complex object; and,

10 passing as a member function parameter to said second proxy on
11 said second machine a reference to said third proxy, in place of said complex
12 object when said complex object is to be a parameter in a member function
13 call to said second object on said second machine.

14 said logic in said second proxy on said second computer to transfer and
15 translate complex data objects comprising the steps of:

16 creating a fourth proxy for said complex object on said first
17 computer which is to reside on said second computer, said fourth proxy

18 containing network linkage and indication necessary to access programmed
19 member functions on said third proxy on said first machine;

20 creating a reference table entry which correlates said fourth proxy
21 to a reference to said third proxy on said third computer, which may be
22 accessed by said fourth proxy to access said third proxy;

23 passing as a member function parameter to said second object
24 from said second proxy on said second computer an indication of said fourth
25 proxy, in place of said reference to said third proxy on said first computer,
26 which represents said complex object on said first computer.

27 said network linkage and indication in said fourth proxy necessary to
28 access programmed member functions on said third proxy on said first
29 computer comprising the steps of:

30 looking up said fourth proxy in said reference table on said second
31 computer to determine which object on said first machine said fourth object is
32 a proxy for, said lookup returning a reference to said third proxy on said first
33 computer;

34 calling the appropriate programmed member functions in said third
35 proxy on said first computer.

36 said linkage and indication in said third proxy necessary to access
37 programmed methods on said complex object comprising the steps of:

38 looking up said third proxy in said reference table on said first
39 computer to determine which object on said first machine said third object is
40 a proxy for, said lookup returning a reference to said complex object on said
41 first computer;

42 calling the appropriate programmed member functions in said
43 complex object.

1 18. A method as claimed in Claim 17 wherein one of said complex objects is
2 said first object on said first computer.

1 19. A method as claimed in Claim 17 wherein said reference table is a
2 database.

1 20. A computer program product for distributing one or more objects of a
2 program across more than one physical device, each object containing one or
3 more programmed member functions, said member functions having complex
4 objects, said complex objects including one or more programmed member
5 functions, as parameters, said computer program product comprising:

6 a computer-readable storage medium have computer-readable program
7 code means embodied in said medium, said computer-readable program code
8 means comprising:

9 computer-readable program code means for identifying all of the
10 objects in the program;

11 computer-readable program code means for determining which of
12 the objects are to reside on a first computer and which of the objects are to
13 reside on a second computer such that the distributed system will consist of at
14 least a first object on a first computer and a second object on a second
15 computer;

16 computer-readable program code means for identifying all
17 programmed methods contained in each object that may be accessed from a
18 remote computer;

19 computer-readable program code means for generating a first
20 proxy and a second proxy for each object that may be accessed from a remote
21 computer, said first proxy residing on said first computer and said second
22 proxy residing on said second computer, said first proxy containing network
23 linkage and indication to access programmed member functions on said
24 second proxy on said second computer including logic to transfer and translate
25 complex objects which reside on said first computer used as member function
26 parameters and said second proxy containing linkage and indication to access

27 said programmed member functions on said second object including logic to
28 transfer and translate complex objects, said complex objects containing one or
29 more programmed member functions and reside on said first computer, used
30 as member function parameters; and,

31 computer-readable program code means for accessing said remote
32 programmed methods through said proxies.

1 21. A computer program product as claimed in Claim 20, wherein:

2 said logic in said first proxy on said first computer to transfer and
3 translate complex data objects comprising the steps of:

4 creating a third proxy, for said complex object, which is to reside on
5 said first computer with said complex object, said third proxy containing
6 linkage and indication to access programmed member functions on said
7 complex object;

8 creating a reference table entry which correlates said third proxy
9 object to said complex object, which may be accessed by said third proxy
10 object to access said complex object; and,

11 passing as a member function parameter to said second proxy on
12 said second machine a reference to said third proxy, in place of said complex

13 object when said complex object is to be a parameter in a member function
14 call to said second object on said second machine.

15 said logic in said second proxy on said second computer to transfer and
16 translate complex data objects comprising the steps of:

17 creating a fourth proxy for said complex object on said first
18 computer which is to reside on said second computer, said fourth proxy
19 containing network linkage and indication necessary to access programmed
20 member functions on said third proxy on said first machine;

21 creating a reference table entry which correlates said fourth proxy
22 to a reference to said third proxy on said third computer, which may be
23 accessed by said fourth proxy to access said third proxy;

24 passing as a member function parameter to said second object
25 from said second proxy on said second computer an indication of said fourth
26 proxy, in place of said reference to said third proxy on said first computer,
27 which represents said complex object on said first computer.

28 said network linkage and indication in said fourth proxy necessary to
29 access programmed member functions on said third proxy on said first
30 computer comprising the steps of:

31 looking up said fourth proxy in said reference table on said second
32 computer to determine which object on said first machine said fourth object is

33 a proxy for, said lookup returning a reference to said third proxy on said first
34 computer;

35 calling the appropriate programmed member functions in said third
36 proxy on said first computer.

37 said linkage and indication in said third proxy necessary to access
38 programmed methods on said complex object comprising the steps of:

39 looking up said third proxy in said reference table on said first
40 computer to determine which object on said first machine said third object is
41 a proxy for, said lookup returning a reference to said complex object on said
42 first computer;

43 calling the appropriate programmed member functions in said
44 complex object.

1 22. A computer program product as claimed in Claim 21 wherein one of said
2 complex objects is said first object on said first computer.

1 23. A computer program product as claimed in Claim 21 wherein said
2 reference table is a database.

1 24. A computer system for distributing one or more objects of a program
2 across more than one physical device, each object containing one or more

3 programmed member functions, said member functions having complex
4 objects, said complex objects including one or more programmed member
5 functions, as parameters, said system comprising:

6 means for identifying all of the objects in the program;

7 means for determining which of the objects are to reside on a first
8 computer and which of the objects are to reside on a second computer such
9 that the distributed system will consist of at least a first object on a first
10 computer and a second object on a second computer;

11 means for identifying all programmed methods contained in each object
12 that may be accessed from a remote computer;

13 means for generating a first proxy and a second proxy for each object
14 that may be accessed from a remote computer, said first proxy residing on said
15 first computer and said second proxy residing on said second computer, said
16 first proxy containing network linkage and indication to access programmed
17 member functions on said second proxy on said second computer including
18 logic to transfer and translate complex objects which reside on said first
19 computer used as member function parameters and said second proxy
20 containing linkage and indication to access said programmed member
21 functions on said second object including logic to transfer and translate
22 complex objects, said complex objects containing one or more programmed

23 member functions and reside on said first computer, used as member function
24 parameters; and,

25 means for accessing said remote programmed methods through said
26 proxies.

1 25. A system claimed in Claim 24, wherein:

2 said logic in said first proxy on said first computer to transfer and
3 translate complex data objects comprising the steps of:

4 creating a third proxy, for said complex object, which is to reside on
5 said first computer with said complex object, said third proxy containing
6 linkage and indication to access programmed member functions on said
7 complex object;

8 creating a reference table entry which correlates said third proxy
9 object to said complex object, which may be accessed by said third proxy
10 object to access said complex object; and,

11 passing as a member function parameter to said second proxy on
12 said second machine a reference to said third proxy, in place of said complex
13 object when said complex object is to be a parameter in a member function
14 call to said second object on said second machine.

15 said logic in said second proxy on said second computer to transfer and
16 translate complex data objects comprising the steps of:

17 creating a fourth proxy for said complex object on said first
18 computer which is to reside on said second computer, said fourth proxy
19 containing network linkage and indication necessary to access programmed
20 member functions on said third proxy on said first machine;

21 creating a reference table entry which correlates said fourth proxy
22 to a reference to said third proxy on said third computer, which may be
23 accessed by said fourth proxy to access said third proxy;

24 passing as a member function parameter to said second object
25 from said second proxy on said second computer an indication of said fourth
26 proxy, in place of said reference to said third proxy on said first computer,
27 which represents said complex object on said first computer.

28 said network linkage and indication in said fourth proxy necessary to
29 access programmed member functions on said third proxy on said first
30 computer comprising the steps of:

31 looking up said fourth proxy in said reference table on said second
32 computer to determine which object on said first machine said fourth object is
33 a proxy for, said lookup returning a reference to said third proxy on said first
34 computer;

35 calling the appropriate programmed member functions in said third
36 proxy on said first computer.

37 said linkage and indication in said third proxy necessary to access
38 programmed methods on said complex object comprising the steps of:

39 looking up said third proxy in said reference table on said first
40 computer to determine which object on said first machine said third object is
41 a proxy for, said lookup returning a reference to said complex object on said
42 first computer;

43 calling the appropriate programmed member functions in said
44 complex object.

1 26. A system as claimed in Claim 25 wherein one of said complex objects is
2 said first object on said first computer.

1 27. A system as claimed in Claim 25 wherein said reference table is a
2 database.